

Hackbuteer1的专栏

走别人没走过的路，让别人有路可走。

目录视图

摘要视图

RSS 订阅

个人资料



hackbuteer1



访问： 3170371次
积分： 30157
等级： **BLOG > 8**
排名： 第116名

原创： 254篇 转载： 8篇
译文： 0篇 评论： 2518条

文章搜索

博客专栏



考研复试上机题
文章： 14篇
阅读： 30745



IT公司笔试题集锦
文章： 19篇
阅读： 483381

文章分类

- C/C++ (63)
- OpenGL (5)
- OSG (1)
- Qt开发 (4)
- 数据结构 (61)
- OGRE (1)
- OGRE编译 (1)
- Ogitor (1)
- 编程之美 (39)
- 面试珠玑 (85)
- 九度OJ (29)
- STL源码剖析 (9)

【公告】博客系统优化升级 【收藏】Html5 精品资源汇集 前端开发人员必须了解的七大技能图谱

C++模板类学习

标签： c++ class overloading system float c

2011-08-31 14:17 25906人阅读 评论(7) 收藏 举报

分类： 编程之美 (38) ▾

版权声明：本文为博主原创文章，未经博主允许不得转载。

1、模板的概念

我们已经学过重载(Overloading)，对重载函数而言,C++的检查机制能通过函数参数的不同及所属类的不同。正确的调用重载函数。例如，为求两个数的最大值，我们定义MAX()函数需要对不同的数据类型分别定义不同重载(Overload)版本。

//函数1.

```
int max(int x,int y)
{ return(x>y)?x:y ; }
```

//函数2.

```
float max( float x,float y)
{ return (x>y)? x:y ; }
```

//函数3.

```
double max(double x,double y)
{ return (x>y)? x:y ; }
```

如果在主函数中，我们分别定义了 char a,b; 那么在执行max(a,b);时 程序就会出错，因为我们没有定义char类型的重载版本。

现在，我们再重新审视上述的max()函数，它们都具有同样的功能，即求两个数的最大值，能否只写一套代码解决这个问题呢？这样就会避免因重载函数定义不全面而带来的调用错误。为解决上述问题C++引入模板机制，**模板定义：模板就是实现代码重用机制的一种工具，它可以实现类型参数化，即把类型定义为参数，从而实现了真正的代码可重用性。模版可以分为两类，一个是函数模版，另外一个类模版。**

2、函数模板的写法

函数模板的一般形式如下：

Template <class或者也可以用typename T>

返回类型 函数名（形参表）

{//函数定义体 }

说明：**template**是一个声明模板的关键字，表示声明一个模板关键字**class**不能省略，如果类型形参多余一个，每个形参前都要加**class** <类型 形参表>可以包含基本数据类型可以包含类类型。

剑指Offer (21)

Spring笔记 (0)

文章存档

2014年09月 (1)

2013年09月 (9)

2013年06月 (1)

2013年05月 (2)

2013年01月 (4)

展开

阅读排行

浅谈C++多态性

(129419)

百度最新面试题集锦

(112329)

C++中的单例模式

(101047)

C++中智能指针的设计和

(72756)

程序员有趣的面试智力题

(65058)

字符串的全排列和组合算

(59260)

链表各类操作详解

(57307)

阿里巴巴笔试题

(55350)

虚函数和纯虚函数的区别

(52754)

N皇后问题的两个最高效

(50121)

评论排行

程序员有趣的面试智力题

(153)

淘宝2011.9.21校园招聘笔

(118)

百度最新面试题集锦

(113)

阿里巴巴笔试题

(112)

浅谈C++多态性

(69)

C++中智能指针的设计和

(65)

一个应届计算机毕业生的

(58)

网易游戏2011.10.15校园

(53)

微软校园招聘笔试题

(51)

C++中的单例模式

(50)

推荐文章

*Android RocooFix 热修复框架

* android6.0源码分析之Camera API2.0下的初始化流程分析

*Android_GestureDetector手势滑动使用

*Android MaterialList源码解析

*Android开源框架Universal-Image-Loader基本介绍及使用

*Android官方开发文档Training系列课程中文版：创建自定义View之View的创建

最新评论

C++中的static关键字

fengzhongpiaoxing: 写的很好，学习了。

虚函数和纯虚函数的区别

eziowayne: @weiaipan1314:那还叫纯虚函数吗

浅谈C++多态性

请看以下程序：

```
[cpp]
01. #include <iostream>
02. using std::cout;
03. using std::endl;
04. //声明一个函数模版,用来比较输入的两个相同数据类型的参数的大小，class也可以被typename代替，
05. //T可以被任何字母或者数字代替。
06. template <class T>
07. T min(T x,T y)
08. {
09.     return(x<y)?x:y;
10. }
11. void main( )
12. {
13.
14.     int n1=2,n2=10;
15.     double d1=1.5,d2=5.6;
16.     cout<< "较小整数:"<<min(n1,n2)<<endl;
17.     cout<< "较小实数:"<<min(d1,d2)<<endl;
18.     system("pause");
19. }
```

程序运行结果：



1、模板类和重载函数一起使用

两者一起使用时，先考虑重载函数，后考虑模板类，如过再找不到，就考虑类型转换，可能会带来精度的变化。

```
[cpp]
01. #include "iostream"
02. using namespace std ;
03.
04. //函数模板
05. template <class T>
06. const T MAX(T a , T b)
07. {
08.     printf("%s\n" , "template") ;
09.     return (a > b) ? a : b ;
10. }
11.
12.
13. int MAX(int x , int y)
14. {
15.     printf("%s\n" , "int int" );
16.     return (x > y) ? x : y ;
17. }
18.
19.
20.
21.
22.
23.
24.
25.
26.
27.
28.
29.
30.
31.
32.
33.
34.
35.
36.
37.
38.
39.
40.
41.
42.
43.
44.
45.
46.
47.
48.
49.
50.
51.
52.
53.
54.
55.
56.
57.
58.
59.
60.
61.
62.
63.
64.
65.
66.
67.
68.
69.
70.
71.
72.
73.
74.
75.
76.
77.
78.
79.
80.
81.
82.
83.
84.
85.
86.
87.
88.
89.
90.
91.
92.
93.
94.
95.
96.
97.
98.
99.
100.
```

playboy_lei: 写的太好了，受教！

链表各类操作详解
编程小王: 楼主你好，我很喜欢你写的博文，就是有图片没有显示出来 能发份原稿给我么？谢谢 772625954@...

链表各类操作详解
alhaocy: 很需要，能不能麻烦您发一份原稿给我呢，一个图都看不到呢！确实很需要！ 1016647213@qq....

浅谈C++多态性
time770880: 感谢分享 收获很大

C++中的static关键字
first_three_sun: 很荣幸充当这第100个受教者，非常感谢。

C语言的可变参数函数的设计
qq_14845089: #define va_arg(ap,type) (*(type *)((ap) += _INT...

程序员面试100题之十六：二叉树
最爱人间这四月天: 情况二那个，当root==a || root==b时。不是应该返回NULL么，根节点没有父节点...

字符串的全排列和组合算法
sdadafsd: 这个..不是数学问题么

```
19. int MAX(char x , int y)
20. {
21.     printf("%s\n" , "char int" );
22.     return (x > y) ? x : y ;
23. }
24.
25. int MAX(int x , char y)
26. {
27.     printf("%s\n" , "int char" );
28.     return (x > y) ? x : y ;
29. }
30.
31. int main(void)
32. {
33.     int a = 3 , b = 5 ;
34.     char x = 'x' ;
35.     double c = 3.4 ;
36.     cout<<MAX(a , b)<<endl ;    //调用重载函数
37.     cout<<MAX(c , b)<<endl ;    //无对应的重载函数，则调用模板
38.
39.     cout<<MAX(a , x)<<endl ;    //重载函数
40.     cout<<MAX(x , a)<<endl ;    //重载函数
41.     cout<<MAX(c , a)<<endl ;
42.     cout<<MAX(a) ;
43.     system("pause") ;
44.     return 0 ;
45. }
```

2、类模板

(1)类模板的具体格式

```
template <class T>
class A
{
}
}
```

在类定义体外定义的成员函数，应该使用函数模板。

```
[cpp]
01. /*
02. 类模板，但是在类外定义成员函数的时候，需要使用函数模板
03. */
04. #include <iostream>
05. using namespace std ;
06. template <class T>
07. class Base
08. {
09. public :
10.     T a ;
11.     Base(T b)
12.     {
13.         a = b ;
14.     }
15.     T getA(){ return a ;} //类内定义
16.     void setA(T c);
17. };
18.
19. template <class T>    //模板在类外的定义
20. void Base<T>::setA(T c)
21. {
22.     a = c ;
23. }
24.
25. int main(void)
26. {
27.     Base <int>b(4);
28.     cout<<b.getA()<<endl;
29.
30.     Base <double> bc(4);
31.     bc.setA(4.3);
32.     cout<<bc.getA()<<endl;
33.     system("pause");
34.     return 0 ;
35. }
```

注意成员函数在类外定义的情况。

3、模板类
主要指的是 STL 模板类

顶 踩
15 1

上一篇 开关和灯泡的对应关系
下一篇 15个Google面试题以及答案~~~你会几个?

我的同类文章

编程之美 (38)

- 程序员面试100题之十五：数.. 2012-06-06 阅读 9155
- 【2012百度之星/初赛下】C... 2012-06-04 阅读 2361
- 【2012百度之星/初赛下】A... 2012-06-04 阅读 3468
- 【2012百度之星/初赛上】B... 2012-06-03 阅读 2278
- 浙江大华2011.10.10校园招... 2011-10-12 阅读 11674
- 亚信联创2011.9.17招聘会笔... 2011-09-26 阅读 15618





- 【2012百度之星/初赛下】D... 2012-06-04 阅读 3885
- 【2012百度之星/初赛下】B... 2012-06-04 阅读 2451
- 【2012百度之星/初赛上】C... 2012-06-03 阅读 2460
- 【2012百度之星/初赛上】A... 2012-06-03 阅读 2607
- 淘宝2011.9.21校园招聘会笔... 2011-09-26 阅读 34809

更多文章

猜你在找

- iOS8-Swift开发教程
- C++语言基础
- C/C++单元测试培训
- 《C语言/C++学习指南》语法篇（从入门到精通）
- 深入浅出MySQL入门必备
- c++模板类学习
- c++模板类学习
- c++模板类学习
- c++模板类学习
- c++模板类学习

查看评论

- 7楼 [qianshanxue11](#) 2016-03-30 14:31发表
 很好
- 6楼 [CrawlWu](#) 2015-08-18 11:56发表
 楼主的一样思考是好的,对我也有帮助,只是代码貌似没有编译一下看看,有点想当然
- 5楼 [景语](#) 2014-10-15 19:34发表
 对于重载函数和模板类的思考是亮点!
- 4楼 [旧时明月](#) 2013-10-10 08:59发表
 谢谢，讲的很好啊。就是想问问在<函数模板的写法>中那个例子，为什么不能将using std::cout; using std::endl;直接用using namespace std代替呢？ 我试了一下，改了之后就错了。。。但是不懂为什么，请教。。。
- 3楼 [测试一下](#) 2013-10-08 22:10发表


```
visual studio 2005\projects\test017\test017\test017.cpp(57) : error C2668: 'MAX' : ambiguous call to overloaded function
visual studio 2005\projects\test017\test017\test017.cpp(23): could be 'int MAX(char,int)'
visual studio 2005\projects\test017\test017\test017.cpp(17): or 'int MAX(int,int)'
1> while trying to match the argument list '(double, int)'
visual studio 2005\projects\test017\test017\test017.cpp(61) : error C2668: 'MAX' : ambiguous call to overloaded function
\visual studio 2005\projects\test017\test017\test017.cpp(23): could be 'int MAX(char,int)'
visual studio 2005\projects\test017\test017\test017.cpp(17): or 'int MAX(int,int)'
1> while trying to match the argument list '(double, int)'
\visual studio 2005\projects\test017\test017\test017.cpp(62) : error C2661: 'MAX' : no overloaded function takes 1 arguments
```

第二个例子编译提示如上

2楼 [quick_isbest](#) 2013-04-15 16:01发表



cout<<MAX(c , b)<<endl ; //无对应的重载函数，则调用模板
这个是不是错了，编译貌似过不了

1楼 [xiaotao258777](#) 2011-09-04 19:45发表



挺好的，真的可以学到

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

- 全部主题
- Hadoop
- AWS
- 移动游戏
- Java
- Android
- iOS
- Swift
- 智能硬件
- Docker
- OpenStack
- VPN
- Spark
- ERP
- IE10
- Eclipse
- CRM
- JavaScript
- 数据库
- Ubuntu
- NFC
- WAP
- jQuery
- BI
- HTML5
- Spring
- Apache
- .NET
- API
- HTML
- SDK
- IIS
- Fedora
- XML
- LBS
- Unity
- Splashtop
- UML
- components
- Windows Mobile
- Rails
- QEMU
- KDE
- Cassandra
- CloudStack
- FTC
- coremail
- OPhone
- CouchBase
- 云计算
- iOS6
- Rackspace
- Web App
- SpringSide
- Maemo
- Compuware
- 大数据
- aptech
- Perl
- Tornado
- Ruby
- Hibernate
- ThinkPHP
- HBase
- Pure
- Solr
- Angular
- Cloud Foundry
- Redis
- Scala
- Django
- Bootstrap